基于软硬件多样性的主动防御技术

仝 青[1], 张 铮[1], 邬江兴[2]

[1]数学工程与先进计算国家重点实验室 郑州 中国 450001
[2]国家数字交换系统工程技术研究中心 郑州 中国 450002

摘要 网络空间的攻击和防御呈现不对称性, 防御往往处于被动地位。基于软硬件多样性的主动防御技术试图改变攻防的不对称性, 是当今网络空间防御技术的研究热点之一。本文介绍了基于多样性的主动防御所利用的主要技术手段, 分析了入侵容忍、移动目标防御和拟态防御三种主动防御框架下的系统实现, 并对比分析了三种主动防御技术的防御效果和优缺点, 最后展望了基于软硬件多样性的主动防御技术的发展方向。

关键词 主动防御; 多样性; 入侵容忍; 移动目标防御; 拟态防御
中图法分类号 TP309 DOI 号 10.19363/j.cnki.cn10-1380/tn.2017.01.001

The Active Defense Technology Based on the Software/Hardware Diversity

TONG Qing[1], ZHANG Zheng[1], WU Jiangxing[2]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[2] National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

Abstract There is an asymmetric phenomenon in the cyber security, where the defense is often passive while the attack is active. The technologies of active defense try to rebalance the situation in the attack-defense game, which has been a question of heat in the research of cyber defense technology. First, the methods based on the software/hardware diversity are introduced, and then three kind of active defense, including intrusion tolerance, moving target defense and mimic defense, are interpreted with several systems and architectures. And then, the defense effect, pros and cons of the active defense are compared and analyzed. Finally, the future work and research direction are discussed.

Key words Active defense; diversity; intrusion tolerance; moving target defense; mimic defense

1 背景

近年来, 随着社会信息化和全球网络化进程的推进, 国家安全和政治、经济、社会发展对网络空间的依赖程度日益加剧, 使得网络空间成为当今社会功能和社会活动的重要支撑。另一方面, 网络空间的广泛脆弱性使世界各国面临前所未有的国家安全形势。网络犯罪、网络恐怖主义、黑客攻击以及网络战对国家安全的威胁凸显, 迫使各国将网络空间安全提升至国家安全的战略高度, 强调网络空间对于国家利益和国家安全的重要地位和重要意义, 将网络空间视为陆、海、空、天之后的“第五空间”。

1.1 网络攻防的不对称性

目前, 网络空间攻防态势基本上处于“易攻难守”状态, 这就造成了攻击和防御的不对称。这种不对称性表现在诸多方面。

数百万行代码的软件系统和数十亿晶体管的芯片系统必须得到完美无缺的设计与实现, 才能做到无漏洞, 这在理论上来说是不可能的, 而针对系统的防御技术则需要保护整个系统的方方面面。从攻击角度, 只需在整个安全链上任意环节找到一个可利用的漏洞, 通过数十行代码就可能破坏或掌控整个系统。

各种自动化攻击工具出现, 极大地降低了攻击

通讯作者: 张铮, 博士, 副教授, Email: ponyzhang@126.com。
本课题得到国家重点研发计划(2016YFB0800104), 上海市科学技术委员会科研计划项目(14DZ1105300), 国家自然科学基金(61572520)和国家自然科学基金创新研究群体项目(61521003)资助。
收稿日期: 2016-09-26; 修改日期: 2016-10-25; 定稿日期: 2016-12-04

技术的难度。密码破解、反编译、漏洞挖掘与渗透等黑客技术也不断发展, 使得攻击者进行网络攻击的节奏明显加快, 从漏洞的发现到攻击的渗透, 所需要的时间间隔越来越短。

现代软硬件规模越来越庞大, 为漏洞和后门提供了天然的掩护。攻击从软攻击发展到软件、硬件和电磁联合攻击, 使得攻击更隐蔽、攻击识别防范更困难。

1.2 网络空间主动防御

在信息安全领域发展的不同阶段, 开发出了一些防御技术和安全产品, 时至今日, 大部分技术与产品仍然在广泛使用, 多属于被动防御技术, 存在一定的防御缺陷。早期的防火墙是一种粗粒度的流量过滤, 对于特定的攻击源有着一定的阻挡作用, 但存在被绕过的可能, 属于静态防御技术。随着技术的融合, 当前的防火墙在一定程度上也成为了通用的入侵检测系统。入侵检测技术是目前最为广泛使用的防御技术之一, 但基于特征的入侵检测存在漏报, 基于异常的入侵检测可能误报, 多数入侵检测系统依然是滞后型的防御, 在分析已知攻击模式的基础上对类似攻击具有检测作用, 却难以有效应对新型攻击手段[1]。防病毒软件对于已知病毒或病毒模式具有发现和防御能力, 但难以识别新型病毒。总体上, 防火墙、防病毒软件、基于特征的入侵检测技术等以阻挡和检测为主要手段, 具有被动性。

在被动防御技术难以应对未知漏洞、后门问题的困境下, 主动防御技术逐步发展并成为研究的焦点。主动防御是指能够在攻击的具体方法和步骤被防御者知悉之前实现防御部署、有效抵抗未知攻击的破坏的防御技术。相对于被动防御技术, 主动防御能够降低攻击对系统的破坏性, 最大程度防范攻击的发生或进行, 尤其是针对未知的攻击, 能够实施更加主动的、前摄的防御。主动防御技术的研究热点在入侵容忍、移动目标、拟态防御、态势感知等方面均有体现。其中, 多数主动防御技术利用了软硬件多样性, 其有效性和安全性在理论和工程层面均得到了较好的论证。在网络空间安全不断发展变化的今天, 基于软硬件多样性的主动防御技术具备较大的发展空间。

本文首先介绍了软硬件多样性的概念和相关技术, 而后主要从入侵容忍、移动目标防御和拟态防御三种技术入手, 对涉及到多样性技术的技术手段和系统架构进行了介绍, 探讨了三种技术的演进过程和继承关系。通过对三种主动防御技术的对比、归纳, 探讨了三种防御技术的优缺点和防御效果, 据此提出基于软硬件多样性的主动防御技术在未来发展中的可能的研究方向和有待解决的问题。

本文剩余部分的组织如下: 第 2 章介绍了软硬件多样性的概念和分类, 例举了主要的多样性技术; 第 3 章依次对入侵容忍、移动目标和拟态防御 3 种主动防御技术的产生、发展和基于多样性的系统、架构、方法进行了介绍; 第 4 章综合对比并讨论了上述三种主动防御技术的优缺点和防御效果。最后对基于软硬件多样性的主动防御技术进行了展望并总结全文。

2 软硬件多样性

软硬件多样性是指功能等价的软硬件在设计、实现、运行的过程中不完全相同的现象[2]。由于多样化的软硬件在某些方面的特异性、异构性, 特定软硬件出现的某些错误在其他软硬件行为中并不会发生, 因而使得软硬件多样性具备容错能力。早在 20 世纪 70 年代, 软硬件多样性技术就已经开始用于信息系统的容灾、备份、容错和可靠性设计等方面。随着计算机系统安全受到越来越多的关注, 由容错技术逐渐形成多种基于软硬件多样性的网络空间主动防御技术, 主要包括基于多样性的入侵容忍、移动目标防御、拟态防御等。

2.1 软硬件多样性分类

依据工程实现上的来源和动机的不同, 多样性大致可分为三种: 自然多样性, 自动化多样性和可控多样性[3]。

“自然多样性”来自于软硬件开发过程, 是在软硬件研发过程中由于受到市场竞争、开发者、执行环境和开发语言等诸多因素的作用, 自然而然形成的一种软硬件多样性形式。现如今被大量使用的各类软硬件存在着大量的自然多样性, 如操作系统, 包含了以 Windows、Linux、Unix 为主流的面向不同平台和应用的百余种操作系统, 数据库管理软件包含了 Oracle, DB2, Sybase, SQL Server, mysql, 以及针对其他各种类型数据库的 Hbase, Cassandra, MongoDB, SequoiaDB 和 Neo4J 等, 硬件设备上则根据厂商的不同、元器件的选择不同有着更广泛的多样性, 另外, 也存在着相同功能通过软件实现和硬件实现两种方式带来的多样性。

自动化多样性通常是指在程序执行过程中通过变量随机化实现的多样性。自动化多样性的产生由管理者控制, 但管理者不参与随机化的过程, 因而无法预料产生的多样性结果。自动化化多样性通常应用在代码层面。静态的随机化可以在源代码层面

或二进制代码层面产生相同程序的不同版本, 典型的技术如代码混淆技术[4]。动态的随机化能够直接或间接地产生同一程序的不同执行过程集, 是在执行环境中实现的, 常用的技术如多样化编译[5]。

可控多样性旨在通过技术方法促成或控制软硬件的多样性表现, 这种多样性常见于多版本软件、开源软件架构和软件产品线。可控多样性相对于自动化多样性和自然多样性而言, 更倾向于人为控制产生的多样性, 典型的可控多样性如多版本程序(*N*-version Programming)[6], 就是在程序功能等价的前提下尽可能产生相对独立的程序版本。可控多样性在可靠性设计方面有着广泛的应用。

自然多样性的产生与可控多样性类似, 但在使用目的上有着较大的差别, 可控多样性比自然多样性对“异构”、“相对独立”的要求更高。

2.2 软硬件多样性技术应用于主动防御

网络空间主动防御技术是相对于传统防御技术提出的, 强调系统在受到攻击前能够主动发现、应对随时可能发生的攻击, 不同于传统的受到攻击后, 通过分析攻击行为特点、病毒特点等方法进行应对的防御技术[8]。因而, 主动防御技术在新型的、未知的攻击行为面前具有较强的抵抗性, 逐渐成为网络安全领域的研究热点。

基于软硬件多样性产生的主动防御技术是主动防御技术的主要组成, 这些主动防御技术以软硬件多样性为基础, 通过冗余设计、动态变迁、表决机制、重配置等技术的组合, 达到增强安全性的目的。

本节围绕多样性, 介绍冗余、表决、动态变迁和重配置技术在主动防御中的特点和应用。

2.2.1 冗余技术

冗余技术出现的时间可以追溯到计算机起源, 早在 1834 年, Lardner 就已经提出“通过多台相同的计算机处理同一运算, 来检测运算结果的正确性, 如果采用不同的运算方法进行计算, 那么结果则更加可靠”[9]。冗余技术在分布式计算以及容错系统中经常使用, 通过构造备份组件或异构组件实现系统的容错、容灾等。

多样性和冗余性在概念上有着较强的相似性, 为便于区分, 本文将“冗余性”的范畴界定为: 2 个或 2 个以上功能等价的模块同时工作, 或者数据的多备份存储。因而, 多样性表示软硬件的一种属性, 而冗余性表示软硬件的一种工作方式或数据的一种存储方式。

软硬件的冗余性既可以通过同构的组件同时工作来实现, 也可以通过异构的组件实现。在数据的容灾、备份需求下, 通常同构冗余更有利于数据迁移和兼容, 而在容错需求下, 异构冗余更有利于避免相同错误同时出现在每个冗余组件或备份数据上。数据的异构冗余可以通过异构的数据结构、存储介质或存储位置等实现, 也可以通过秘密共享达成数据信息的冗余。

2.2.2 表决

表决机制与冗余技术是紧密关联的, 通过表决可以从多个冗余组件的输出结果中得到相对正确的输出。对于同一个输入, 多个冗余组件的响应结果应当是等价的。然而在现实网络环境中, 组件随时可能受到攻击或恶意破坏, 也可能发生功能异常, 从而导致输出异常, 使冗余部件的响应结果出现不一致现象。表决技术就是用来比较冗余响应的差别, 并给出一个相对正确的响应结果作为系统的输出。

表决包括两个关键步骤: 第一步是比较响应, 第二步是达成一致结果。根据比较对象的不同, 比较算法包括针对文本、图像、音频、视频等的多种算法。根据比较的精度有粗略比较和精确比较之分。比较算法中常用到的矩阵包括编辑距离矩阵和哈希矩阵。编辑距离[10]通常用于需要权衡数据修改代价的数据比较算法中。哈希矩阵则用于大数据流的快速比较, 需要使用 MD5, SHA 等加密算法。表决的第二步需要根据比较的结果达成一致输出。常用的表决算法包括全体一致表决算法, 大数表决算法, 中值表决算法等[11]。具体到不同的应用场景, 表决算法可以扩展, 如最大近似表决, 基于历史信息的带权表决[12]。表决算法的效率在主动防御技术的实现中要求较高, 尤其在面对大流量、高访问频率的情况下, 需要在效率与准确性之间权衡。在解决拜占庭问题中, 使用的拜占庭表决算法也是一种重要的表决技术, 应用于分布式多节点通信环境下, 可以达到容错和系统容侵的目的[15]。

2.2.3 动态变迁

动态变迁使系统具有动态性, 打破了长期以来的静态设计[16], 常见的方法如轮转服务, 负载均衡, 流量迁移等, 都是通过将服务、数据在服务过程中通过合理分配实现迁移, 达到系统性能最佳或安全性最强的目的。动态变迁是即时的调度过程, 评估系统的状态, 对资源进行优化调度, 如通过负载均衡提高系统的效率, 通过轮转服务替换故障的服务器或选择性能更优的服务器进行工作。动态变迁也有随机变迁的策略, 在多样的软硬件中, 将流量或服务随机迁移到某一运行环境上, 使用户无法确定服务来自于软硬件池中的哪一个个体, 从而达到保护系

统信息不被利用的目的。同构组件之间的动态变迁对系统的可持续工作能力提供一定的保证, 而异构组件之间的动态变迁不仅能够提供持续可用性, 同时具备了一定的防御能力。

动态迁移通常利用异构的软硬件实现, 一方面避免相同故障的再次发生, 另一方面, 利用异构性实现功能、性能以及组合策略等的优化, 在入侵容忍系统设计方案和移动目标防御中应用动态变迁技术的系统[[17]-[20]]多利用异构组件实现。

2.2.4 重配置

重配置是指系统通过一定方式达到某个新状态, 新状态与先前的任何状态没有关联或关联度低, 从而消除系统在先前状态下受到的影响[21]。常见的重配置方法如回滚、鱼缸策略、清洗恢复和运行参数重配置等[22]。重配置是在一个系统内部通过改变软硬件参数等方式以达到系统的更新, 起到恢复的作用, 其多样性是通过系统自身状态的改变而实现的。重配置技术既可以是前摄式的, 在系统未受到攻击时, 就进行重配置, 以保证系统的动态性; 也可以是反馈式的, 在发生攻击或故障以后, 通过重配置清除攻击造成的故障, 并使系统达到更安全的状态。

重配置技术属于可控多样性的一种实现方式, 同时也是在时间维度上实现的多样性。通过回滚和清洗操作, 可以在一定程度上保证系统的无间断服务, 即高可用性。通过重配置运行参数, 可以实现不同时间下不同的运行环境, 使同一系统产生时间维度上的多样性。

重配置技术与动态变迁技术同属于动态性技术, 动态变迁对冗余资源的需求较高, 而重配置技术能够在较少的资源条件下实现多样性, 资源代价相对较低。然而, 在实现主动防御方面, 两者都需要有较高的执行效率, 才能实现较好的防御效果。

3 基于软硬件多样性的主动防御技术

网络空间主动防御是相对于被动防御提出的, 传统的防御手段如入侵检测、病毒检测、防火墙等方法往往是在攻击发生以后, 通过分析攻击行为特征、病毒代码等提出针对性的防御措施, 并辅以沙箱、蜜罐等手段捕捉攻击行为, 通过打补丁、软件升级等方式试图减少软硬件漏洞[8]。然而这些手段难以从根本上消除漏洞, 为系统提供安全的运行环境, 属于滞后型的防御手段。主动防御技术目标通常是构造安全的系统架构, 或运行方式, 尽可能从根本上增大攻击的难度, 降低攻击成功的概率, 对广义的攻击行为进行有效的遏制, 从而实现系统的安全。

入侵容忍, 移动目标防御和拟态防御是三种典型的主动防御技术。受限于技术产生的时代背景, 三种防御技术在网络空间防御中发挥的作用不尽相同。

需要说明的是, 入侵容忍、移动目标和拟态防御三种技术体系不仅包含了多样性技术, 也包含了其他不依赖多样性的实现技术, 多样性技术作为一种典型技术被用于设计系统和架构以实现特定的安全性或可用性目标。本文的讨论主要在多样性相关的技术领域内展开, 尤其是对入侵容忍的分析, 以基于多样性的技术和系统为主, 以便于横向的比较和分析。

3.1 入侵容忍

3.1.1 入侵容忍产生背景

入侵容忍技术是由容错技术发展而来的。从计算机出现开始, 计算机系统的错误处理技术就同步发展起来。容错技术最初针对的是计算机系统, 尤其是分布式系统的计算结果一致性问题提出的。20 世纪 80 年代, 容错技术开始应用于恶意漏洞的防御, 由“容错”发展为“容侵”, 从此逐渐产生入侵容忍的概念[23]。入侵容忍借用容错技术来达到容忍入侵、维持信息系统正确服务的目的, 该技术也在很大程度上保护了信息系统的可用性, 是当时主流的信息系统安全技术之一。

基于入侵容忍技术, 产生的系统被称为入侵容忍系统(Intrusion tolerant system, ITS)。入侵容忍系统没有明确的和广泛采用的定义, 一种概括[5]为: 一种系统, 即使在面临部分组件被成功攻击时, 仍然可以持续正确地工作且向用户提供预期的服务。根据其他相关文献[24]对入侵容忍系统的解释, 与上述说法基本一致。入侵容忍出现在主动防御提出之前, 是一种早期的主动防御技术, 因而在设计思路上有多种方向, 部分系统的设计和实现与移动目标的防御思路相似, 具体异同点在 3.2 节中分析。

3.1.2 入侵容忍系统

在入侵容忍技术的发展过程中, 有不少入侵容忍系统架构被提出。在美国国防部高级研究计划署(DARPA)的支持下, 出现了若干个具有代表性的入侵容忍系统。欧洲也提出了入侵容忍项目研究计划。在短时间内, 形成了一批入侵容忍模型、架构、系统和组件等, 其中基于多样性的入侵容忍系统是典型代表。

SITAR(Salable Intrusion-tolerant Architecture for Distributed Services)[27]是为分布式服务设计的入侵容忍架构, 该架构利用多样性构建服务器池, 服务请求被分发至不同的服务器上进行处理, 并通过表

决输出唯一响应, 以此达到容忍入侵的目的。除此之外, SITAR 使用了入侵检测技术对每台服务器的响应进行规则匹配, 以发现异常; SITAR 同时具备有重配置模块, 收集架构内其他模块的入侵警报信息, 评估威胁态势, 进而改变系统配置以改变安全防御等级。

SITAR 系统从技术上具备了主动防御技术的大部分特点, 包括冗余、重配置等, 然而技术的组合缺乏精简, 在防御思路上仍然没有跳出入侵检测的桎梏, 因而不可避免地会存在效率低下, 服务响应时延大的缺点。从防御效果上看, SITAR 侧重于保证系统的可用性。

欧洲 MAFTIA(Malicious- and Accidental-Fault Tolerance for Internet Applications)项目[23]设计了一系列理论模型、协议和机制, 构建了一个自下而上的全新的系统架构来达成入侵容忍, 除了通信协议和架构的全新设计所带来的安全性以外, 在实现入侵容忍方面, 采用了冗余技术和拜占庭表决的方法, 对冗余组件的处理结果进行表决从而检测出错误结果。由于 MAFTIA 的架构设计与通用架构的兼容性较弱, 配套的协议虽然具有较高的安全性预期, 却难以推广应用。

在协议安全、数据安全、通信安全等方面, 多数系统和组件的设计采用了拜占庭表决、秘密共享、门限机制等方法, 实现多节点通信安全、数据库系统安全、协议安全、认证安全等。该类入侵容忍系统本质上不通过多样性实现入侵容忍, 主要依赖于拜占庭表决算法解决拜占庭问题[28], 或利用秘密共享和门限机制保证数据的安全性。

COCA(Cornell Online Certification Authority)[29]通过拜占庭表决实现高可用性, 结合主动恢复机制抵抗部分攻击对系统的破坏性。ITDOS(Intrusion Tolerant Distributed Object Systems)[30]基于拜占庭表决方法研究了一个多播协议组件, 以保证分布式系统的通信可靠性。OASIS(Organically Assured & Survivable Information Systems) [31]使用了门限机制和秘密共享[[32]-[34]]实现具有容忍能力的存储系统。秘密共享实现了数据的冗余, 应用于数据安全管理中可以防止单点失效, 抵抗部分副本的损坏, 同时还可以增强数据的机密性。CODEX (COrnell Data EXchange) [31], ITTC(Intrusion Tolerance via Threshold Cryptography)[34]使用了类似的安全方案, 对数据的可用性、正确性以及机密性进行了保护。

上述方案的共性是利用了冗余技术, 无论拜占庭表决, 还是秘密共享, 均起到了利用冗余增强可用性的作用, 特别的是秘密共享与门限机制的组合兼顾了可用性、正确性和机密性。

ITUA (Intrusion Tolerance by Unpredictable Adaptation)[17]实现了两方面的入侵容忍, 在组通信协议设计上, 采用了拜占庭表决, 保证组播通信的可靠性。在服务器系统的安全上, 当检测到攻击或失效的服务器时, 改变资源配置, 将访问流量迁移至其他服务器上, 尽可能保证系统正常服务状态。通过检测的方式来发现攻击和失效服务器, 这种变迁信号的来源依赖于入侵检测, 从一定程度上仍然属于被动的防御方式。

ITUA 在组通信协议中采用了拜占庭表决, 然而在服务器系统安全的设计中没有采用表决的方式, 而是利用了动态迁移技术, 类似的系统在入侵容忍系统的发展过程中也占了相当的比例。

ITSI(Intrusion Tolerant Server Infrastructure)[35]通过负载均衡技术在服务器组内迁移和分发流量, 每台服务器都配备了入侵检测系统, 用于检测攻击的发生和系统的失效, 通过中心控制器控制失效服务器上的流量迁移。

FOREVER(Fault/intrusiOn REmoVal through Evolution & Recovery)系统[36]由多个服务器组成, 单台服务器工作, 当工作状态的服务器受到攻击或发生故障时, 服务流量就会被迁移到其他服务器上, 而受到攻击的服务器则离线, 分析系统受损原因和结果, 并进行重配置, 使服务器恢复到安全状态或提升安全等级; 同时 FOREVER 系统在未受到攻击时也会周期性地轮转服务器上线工作, 因而该系统既利用了主动防御, 同时也沿袭了被动防御。

3.1.3 小结

入侵容忍的发展带动了一批主动防御技术的应用, 促成了主动防御思想的萌芽。

从入侵容忍系统设计、应用的特点可以看出, 入侵容忍系统的主要目标是可用性、可靠性或称“可生存性[37]”。以表决技术为主体的入侵容忍系统偏重于系统的可用性保障; 少数系统以服务器作为冗余对象, 对请求响应流进行表决, 实现服务器系统的高可用性。以动态迁移和重配置为主体的入侵容忍系统多以服务器为对象, 动态迁移利用的是自然多样性, 重配置实现可控多样性, 达到维持系统正常服务的目的, 这种入侵容忍系统的实现思路与移动目标防御有着较大的相似性, 然而在防御目的上存在一定的差别, 具体讨论见 3.2.3。

入侵容忍系统的设计在可靠性设计上有较大的价值, 然而也面临着代价高昂的问题, 多数系统采用了虚拟化技术和面向分布式系统的方式开展研究,

在入侵容忍系统的代价、性能等方面的考量工作有所欠缺。在主动防御方面, 出现了以冗余、表决、动态变迁和重配置为主的主动防御技术, 在应用上也存在由容错、容侵向主动防御方向发展的趋势。

3.2 移动目标防御

3.2.1 移动目标防御产生背景

为破解网络防御困局, 以美国为首的部分国家积极转变防御理念, 以主动防范未知漏洞或威胁为目标, 以大幅度增加网络攻击风险和代价为手段, 着力增强网络防御的灵活适应性与动态自主性, 大力寻求“改变游戏规则”的新技术, 以理论和技术的革命性创新确保美国在网络攻防领域的压倒性优势, 并制定了一系列的战略规划、计划和纲领性文件, 开展顶层设计。2011 年美国科学技术委员会发布了《可信网络空间: 联邦网络空间安全研发战略规划》, 将“移动目标防御 MTD(Moving Target Defense)”确定为“改变游戏规则”的革命性防御技术[38], 并制定了由联邦政府、产业和学术机构共同参与的网络安全研发框架, 保障研发规划的落实。

在美国带动刺激下, 俄、英、法、印、日、德、韩等国纷纷跟进, 将网络空间安全提升至国家战略层面, 全面推进相关制度创设、力量创建和技术创新, 试图在塑造全球网络空间新格局进程中抢占有利位置。以“主动变化、增加攻击者攻击难度”为典型特征的主动防御技术成为了网络防御技术发展前沿。

移动目标防御旨在设计能够可靠地工作在非安全环境中的弹性系统, 其技术愿景是在多个不同的系统维度上, 开发、分析和部署防御者可控的、随时间动态迁移和变化的机制和策略, 以限制自身脆弱性的暴露, 降低被攻击机会, 同时大幅增加攻击者的成本[39]。

移动目标防御技术能够增大攻击者探测系统信息和发起攻击的难度, 然而却需要频繁地改变系统配置, 对系统性能而言是一个不可忽略的损失。

3.2.2 移动目标防御应用

移动目标防御在计算机系统的多个层面具有应用空间。

网络层面的移动目标防御, 达到的目标是隐蔽用户以起到保护作用。

MT6D (Moving Target IPv6 Defense)[40]针对 IPv6 地址空间下的通信保密问题提出了一种移动目标防防御方法。MT6D 采用了基于密钥的动态地址变换算法, 建立会话后, 接收方和发送方的通信 IP 地址将会不断地变换, 而对方的地址是由双方通过密钥和变化时间间隔计算得到的, 因而不需要在通信的过程中额外增加流量用于沟通变换后的地址, 同时也避免了通信过程中的第三方攻击。通过频繁地改变网络中消息发送方和接收方的 IP 地址, 来防止攻击者发现通信双方的真实身份, 从而阻止针对特定节点的网络攻击, 如数据包劫持。这种动态地址变换利用的是 IP 地址的重配置产生的可控多样性, 在网络攻击的扫描探测阶段有着较好的防御效果, 然而对于公开目标, 如 web 服务器, 则难以发挥作用。

类似的系统还包括 MUTE(Mutable Network)[41], 支持网络配置(如 IP 地址、端口号等)的动态随机变化, 使攻击者无法探测系统信息, 以应对扫描蠕虫、侦查、指纹攻击等; OF-RHM(OpenFlow Random Host Mutation)[42], 基于软件定义网络的 OpenFlow 协议, 通过 OpenFlow 控制器建立和改变各通信节点的实际 IP 与虚拟 IP 的映射关系, 从而大量减少实际 IP 在通信过程中的暴露机率, 有效防御各类扫描攻击。

动态平台的设计主要利用平台的自然多样性实现。

MAS(Moving Attack Surface)[18]通过使用不同软件组合配置的虚拟服务器栈, 形成了多样化的攻击面。MAS 方法在固定轮换调度策略或事件驱动的基础上, 设计了一套采用多样化的脱机服务器轮转代替一组在线服务器的轮转策略。由于在线 VS 的组合是随机改变的, 攻击者将不得不面对多个不断变化且不可预知的攻击面。

SCIT(Self-Cleansing Intrusion Tolerance)[19]框架被应用到了 DNS 和 web 服务器上来设计入侵容忍系统。SCIT 轮转纯净的(未受到攻击或感染的)服务器上线服务, 而下线的服务器则接受清洗操作, 恢复到纯净的状态, 达到限制服务器的在线暴露时间的目的, 从而减小攻击者挖掘利用漏洞的可能性。SCIT 同时上线的有三台服务器, 这三台服务器可以是不同功能的服务器, 也可以是用于均衡负载的相同功能的服务器, 总体上起到了分担流量压力的作用。

MAS 与 SCIT 有着很大的相似性, 因为它们都使用虚拟化、VS 映像的复制、恢复以及轮转等技术, 不同的是 SCIT 对虚拟服务器的多样性要求不高, 而 MAS 明确提出虚拟服务器应是多样的, 尤其是在每一次轮换发生前后, 虚拟服务器组应保证一定的异构性, 以确保攻击面的改变。

虽然 SCIT 作为入侵容忍系统被提出, 然而在设计和实现目的上都更贴近移动目标防御的设计思想, 多篇文献[43-46]对 SCIT 的引用也将其归为移动目标防御系统, 因而本文将 SCIT 列入移动目标防御技术体系下进行介绍。SCIT 系统作为一个特例, 表明了入侵容忍技术与移动目标防御技术之间一定的关联性。

TALENT(Trusted Dynamic Logical Heterogeneity System)[20]针对内核级、操作系统层面和硬件层面的防御进行了移动目标防御设计, 以虚拟化专用服务器作为容器, 应用运行在容器中, 下层的操作系统和硬件资源, 如文件系统、内存、套接字、设备等, 均进行虚拟化。通过迁移应用所在的容器, 达到改变应用运行环境的目的。

文献[47]在云环境下利用多样化的虚拟机池, 通过快照和恢复技术, 实现流量和服务的动态迁移。

除了网络层和平台层, 移动目标防御在运行环境、软件层和数据层也存在着广泛的应用[39]。如运行环境层的地址空间随机化技术(address space randomization, ASR)和指令集随机化技术(instruction set randomization, ISR)[48]能够防御攻击者挖掘软件漏洞, 其中 ASR 技术在学术和商用领域都已经趋于成熟, 应用较为广泛。

3.2.3 小结

移动目标防御主要利用了动态性技术, 如动态迁移、重配置, 防御的目标是改变系统的静态属性, 通过不可预测的前摄的变化, 达到迷惑攻击者, 隐蔽脆弱性的目的。

移动目标防御的思想与部分入侵容忍系统的设计非常相近, 然而在防御目的上, 移动目标则具有更明确的主动防御特征, 强调前摄式、不可预测的变化, 最大程度地避免攻击作用于目标; 而入侵容忍系统设计中, 动态性技术多用于受攻击系统的修复和服务的维持, 因而, 在不同的安全目标下, 入侵容忍和移动目标对技术的组合、优化处理有着不同的方向。

移动目标防御技术虽然能够有效地实现动态性, 然而频繁的变化对系统的性能可能造成一定的影响[49]。另外, 移动目标防御的变迁多为随机变迁, 随机变迁是否能够有效地隐蔽漏洞、对多样性的需求量大小都是在实际应用中关系到系统效能的问题。

移动目标防御的理论和实践经过十余年的努力, 已日趋成熟, 能够在多层面提供安全防御能力。近年来, 虚拟化技术、多核技术、软件定义网络等的发展为移动目标防御的实现节约了较大的成本, IPv6、云计算和云安全的发展, 为移动目标防御提供了广阔的应用空间。

3.3 拟态防御

3.3.1 拟态防御产生背景

近年来, 伴随着多起震惊世界的网络安全事件的曝光, 全球各国对网络空间安全的重视程度不断提高。在国家级网络空间安全战略的引领下, 拟态防御针对网络空间未知漏洞和后门问题, 提出采用“有毒带菌”组件和“沙滩建楼”方法构建由内生机理保证的风险可控、安全可信的系统[8]。拟态防御在网络层面、系统层面、数据层面等多层次进行基于异构性的主动防御技术研究, 形成了相关的拟态防御理论和技术[50-54]。

3.3.2 拟态防御架构与系统

拟态防御以软硬件多样性为基础, 以异构性最大化为主要目标, 综合了冗余、表决、主动重构、动态迁移、重配置等技术, 构建了动态异构冗余结构(Dynamic Heterogeneous Redundancy, DHR)[8], 如图 1 所示。

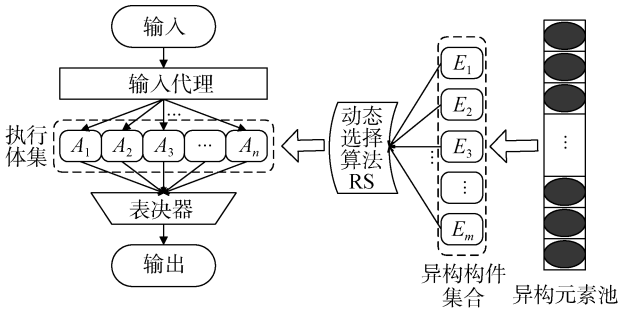

图 1 DHR 结构

拟态防御通过破坏网络攻击对被攻击目标的环境依赖, 扰乱攻击信息链, 从而减少未知漏洞和后门的利用度, 增加网络攻击难度和代价, 降低网络空间安全风险。

动态异构冗余结构由输入代理、异构元素池、异构构件集、动态选择算法、执行体集和表决器组成, 如图 1 所示。系统设计软硬件模块可配置的异构元素池, 由异构元素组建 m 个功能等价的异构构件, 按动态选择算法 RS 动态选出 n 个异构构件作为一个执行体集$(A_1, A_2, \cdots, A_n)$。在任意时刻, 系统输入代理将输入转发给该执行体集中的各执行体, 这些执行体运行后的输出结果提交给表决器进行表决, 得到系统输出。

DHR 结构是拟态防御系统的一种体系结构, 其核心是通过对异构执行体的处理结果进行表决, 得到相对正确的输出。在防御效果上, 拟态防御依靠表决输出的方式, 扰乱传递至攻击者的攻击效果信息, 从而模糊攻击者对攻击结果的判断, 进而阻断攻击链; 同时表决技术本身为系统的高可用性提供保障。

异构性最大化是拟态防御追求的主要目标, 最大化的异构性能够尽可能减小表决出错误结果的可能。主动重构技术是一种动态性技术, 利用组合优势, 以较少的资源代价产生较丰富的多样性, 是拟态防

御多样性的基本来源。其原理是根据不同的任务或作业、资源配置状况、服务质量要求、处理负荷、运行效能等因素, 选择或定制相对理想的计算环境, 包括计算和控制部件、存储部件、互联部件、输入输出部件等。主动重构技术是拟态计算研究的关键技术[52], 主要用于高效能计算研究, 而用于拟态防御技术研究中, 能够根据异构性需求, 组合符合要求的构件搭建异构冗余资源池。动态选择和变换执行体集在时间维度上进一步增大了系统的异构性, 同时具备了移动目标防御的防御效果。总体而言, 拟态防御技术试图以较小的资源开销获取较高的安全性, 在主动防御技术发展中, 性价比相对较高。

DHR 结构, 拟态防御在路由器和 web 服务器上进行了系统设计与验证。

拟态防御路由器[53]在数据层、控制层和管理层引入冗余性、异构性和动态性, 支持 OSPF 和 BGP 两种典型域内/域间路由协议的并行执行和表决输出, 从原理上实现了 DHR 结构。拟态防御路由器结构如图 2 所示。

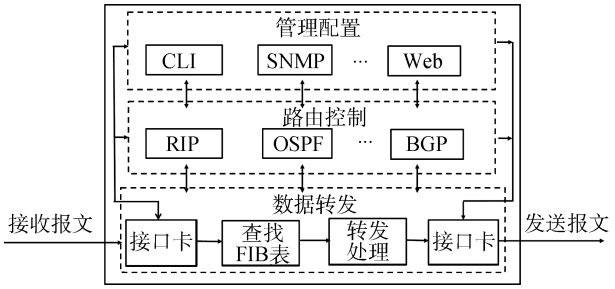

图 2 拟态防御路由器结构[53]

路由器功能结构包括三个平面, 即数据转发平面、路由控制平面和管理配置平面。数据转发平面包括查表和转发引擎, 负责对到达路由器的数据进行查表转发; 路由控制平面包含 OSPF、BGP、RIP 等路由协议, 负责基于路由消息进行路由计算, 生成转发平面所需的转发表项; 管理配置平面包括命令行、SNMP 等协议, 负责系统的配置管理与维护。

具体实现上, 在管理配置平面引入多个管理执行体, 在路由控制平面引入多个路由计算执行体, 在数据转发平面引入多个数据负载语义变换执行体, 从而在各个平面构建出异构冗余的处理单元。同时引入输入代理进行消息分发, 引入拟态裁决进行输出结果裁决, 通过动态调度实现各个平面执行的动态化。

拟态防御 web 服务器[54]基于 web 服务器的层次化结构, 在系统层、虚拟化层、服务器软件层、应用脚本层和数据层等多个层次上实现了 DHR 结构。基于 web 服务器“请求-响应”的服务特性, 利用自然多样性和可控多样性作为异构的基础, 通过动态选择算法实现执行体集的异构性最大化。拟态防御 web 服务器结构如图 3 所示。

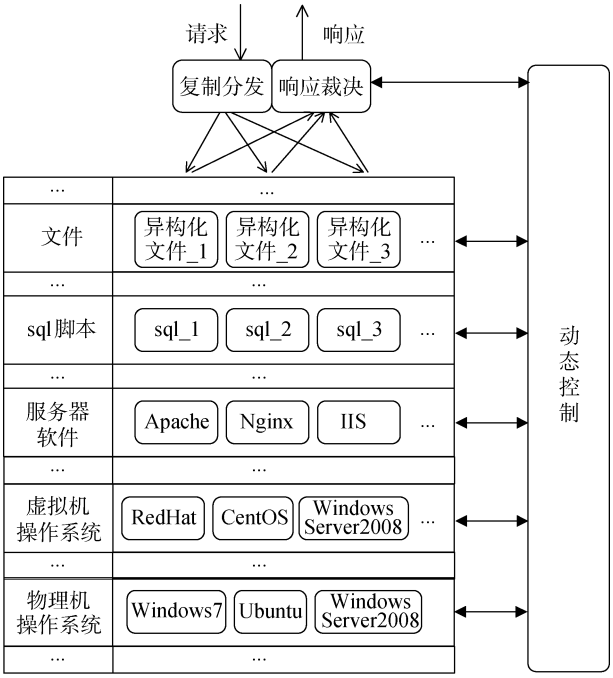

图 3 拟态防御 web 服务器结构[54]

拟态防御 web 服务器主要在文件层、sql 脚本层、服务器软件层、虚拟机操作系统层以及物理机操作系统层 5 层实现拟态防御模型。每一层由多种可用的异构化的构件组成, 通过主动重构构成多样的服务器软件栈, 从而生成异构的服务器执行体。在服务入口和出口处, 设置了请求分发和响应裁决模块, 通过表决算法得到唯一的响应输出, 同时将不一致信息反馈至动态控制模块。另外, 在服务器与数据库之间进行对异构 sql 语句的表决, 以保证数据库接收到的查询语句的正确性、合法性。

除了路由器和 web 服务器的设计, 拟态防御在网络存储、数据库和云环境等应用场景下也部署了研究计划, 同时拟态防御技术和组件逐步迈向商用环境, 寻求更多的应用场景。

3.3.3 小结

在技术层面, 拟态防御主要利用的是冗余、表决、主动重构和动态迁移技术, 在数据层、代码层也利用了重配置技术, 综合了多种主动防御技术。

拟态防御利用的软硬件多样性侧重于“异构性”。拟态防御技术依赖冗余和异构, 利用动态性技术扩充多样性。同时, 组件级安全性的提高和被动防御技术的辅助能够为拟态防御架构带来更高的安全性。

拟态防御对被动防御技术(防火墙、入侵检测等)的依赖性大幅降低, 但仍然保持对被动防御的兼容性。另外, 拟态防御在 DHR 基本结构的设计中通过主动重构思想节约了成本, 相对于入侵容忍和移动目标, 在设计上具有较高的性价比。

拟态防御通过异构性最大化达到了扰乱攻击信息链的目的, 使攻击者难以正确判断系统指纹、已发起攻击的执行结果, 因而在攻击链的各个环节上对攻击造成阻碍, 延长了攻击周期, 大幅提高攻击难度、降低攻击成功概率, 从而实现系统的相对安全。

拟态防御技术是国内网络空间主动防御领域的代表技术之一。面临日益严峻的网络攻防形势, 拟态防御在在新的技术背景下针对当前网络威胁和未来的防御技术发展进行了研究部署, 其研究价值和实际意义不可小觑。

4 对比分析

入侵容忍、移动目标防御和拟态防御在不同的计算机安全发展环境下产生和发展, 对当时的网络空间安全发挥了一定的作用。三种主动防御技术有着诸多异同点, 表 1 列举了三种主动防御中基于多样性的系统的特点以作对比分析。

表 1 现有工作对比

主动防御技术	系统、架构	自然多样性	自动化多样性	可控多样性	冗余	表决	动态迁移	重配置	防御效果	优点	不足
基于多样性的入侵容忍	SITAR	√		√	√	√		√	削弱攻击的破坏力	能够应对系统或数据部分节点受攻击或失效的问题, 维持正常服务, 增强系统的可用性和数据的安全性	冗余带来的成本代价较高
	MAFTIA		√		√	√					
	ITUA	√	√				√	√			
	FOREVER	√		√			√	√			
移动目标防御	MT6D			√				√	增大发起攻击的难度	在不同层面的应用能够起到一定的隐蔽目标的作用, 保证系统的安全性和特定应用中的保密性	轮转调度的周期性规律容易被掌握; 系统的多样化呈现, 有可能起到增大攻击面的反作用; 频繁的变化可能造成系统性能损失
	SCIT	√	√		√		√	√			
	MAS	√	√				√	√			
	TALENT	√		√			√	√			
	云环境下的实现	√					√	√			
拟态防御	拟态防御路由器	√			√	√		√	扰乱攻击信息链, 增大攻击发起和持续进行的难度	以较高的性价比保证可用性、机密性和完整性	技术成熟度有待进一步提高; 控制管理核心的安全性依赖于一定的被动防御技术
	拟态防御 web 服务器	√		√	√	√	√	√			

从三种技术所包含的系统、架构分析, 基于多样性的入侵容忍系统基本分为两大类: 以冗余表决为技术主体的入侵容忍和以动态性技术为主体的入侵容忍, 而两种入侵容忍的共同目标是保证可用性, 即当系统受到破坏时能够继续维持正常服务或在最短时间内切换服务器使服务持续, 尽可能减小平均故障时间。因而, 在软硬件多样性的使用上, 基于多样性的入侵容忍系统集中在自然多样性的利用上, 并不强调异构性, 更多的是利用系统构件或组件特性最直接的多样性来源, 如服务器的入侵容忍设计中利用服务器的自然多样性实现; 涉及时间维度的变迁时, 服务器采用自动化多样性。

移动目标防御的系统设计主要利用动态性技术实现, 其利用方式与入侵容忍不完全相同。移动目标防御的原理是通过变化造成攻击者的迷惑, 使其难以定位攻击目标, 因而强调的是系统的动态性带来的不确定性, 不同于入侵容忍单纯的“容忍性”, 但移动目标的变化能够同时达到“不确定性”和“容忍”两种防御效果。移动目标在不同层面上可以根据不同应用的特点选择不同的动态性技术和多样性实现, 网络层的移动目标防御采用可控多样性, 平台层面侧重于自然多样性, 代码层面的移动目标侧重于自动化多样性。在多样性的控制上, 移动目标防御对异构性有本质的需求, 而在实现中则通常默认“多样性”等同于“异构性”, 即忽略了相同软件相近版本的异构性与不同软件的异构性相对大小。

拟态防御技术由于发展时间较短, 成型的系统较少, 从已有架构中可以发现, 拟态防御综合利用了多种主动防御技术, 不是简单的叠加, 而是以异构性为核心, 组合各项技术起到扩充异构性的作用。技术上, 冗余、表决、动态迁移和重配置都有涉及, 在防御原理上则是通过扩大异构性, 通过表决产生

相对正确的响应, 向攻击者呈现非预期的攻击响应, 从而达到扰乱攻击信息链, 阻断攻击链的效果。在多样性的选择上, 拟态防御以异构性最大化为目标, 增大了对可控多样性的利用。自然多样性通过动态选择算法的选择, 在前端冗余呈现上形成可控多样性。因而可以说拟态防御主要依赖可控多样性。

从三种技术的防御效果来看, 基于多样性的入侵容忍达到了削弱攻击破坏力的效果。该技术以维持系统可用性为主要目的, 使系统具有较强的生存能力和恢复能力, 从而减小平均故障时间, 提高系统可生存性, 保证了服务、数据等的可靠性。另外入侵容忍对于性价比的讨论较少, 直观上看, 冗余和表决将带来较高的资源成本和时延, 因而性价比问题可能是导致入侵容忍技术没落的主要原因。

移动目标防御能够提高攻击门槛, 对攻击目标起到一定的隐蔽作用。由于攻击行为一般具有较强的针对性, 往往针对静态的系统特性设计、发起攻击, 因而, 通过动态变化使系统静态性减弱, 能够使攻击者难以定位攻击目标, 从而起到一定的隐蔽作用, 增大了攻击发起的难度。然而, 要保持动态性和有效的防御, 需要系统具有较高的变化频率, 可能对系统的性能会造成一定的损失, 性能和变化频率的折衷将是移动目标防御研究的重点之一。另一方面, 多样呈现的系统在特定时刻下仍是单一性质的系统, 可能会给攻击者提供更大的攻击面和更多的攻击目标, 对系统的防御起到反作用。

拟态防御能够扰乱攻击者与被攻击对象之间的信息链, 扰乱攻击者的判断, 从而造成攻击发起难、持续难、再现难。拟态防御既能够维持可用性, 也能够对被攻击目标起到隐蔽作用。与移动目标的隐蔽原理不同, 拟态防御通过表决输出的方式“中和”或掩盖被攻击目标的输出, 从而对外表现为无异常或攻击无效, 扰乱攻击者对攻击成败和效果的判断。拟态防御的技术组合具有调优的潜力, 能够利用相对较少的资源开销实现相对较高的防御能力, 具有较好的发展前景。未来工作中需要技术进一步成熟, 同时对于拟态防御架构中的可信根的防护问题需要更多的研究投入。

除了以上异同点, 还需要关注三种防御技术的单点失效问题。由于主动防御技术的目的不在于设计完美无缺的系统, 且漏洞具有不可避免性, 因而绝对安全的系统是不存在的。单点失效问题是普遍存在的, 且目前已有较为成熟的解决方法, 如容错技术的焦点即是解决单点失效问题, 也因此, 继承于容错技术的入侵容忍技术在特定的系统实现中能够较好地避免单点失效。相比之下, 移动目标防御和拟态防御均存在调度、管理等需求, 因而与此相关的控制模块的设计可能导致单点失效问题, 如移动目标的动态变迁管理、拟态防御的输入代理、输出代理等, 均有可能导致单点失效。然而, 拟态防御的动态异构冗余模型并未限定输入、输出代理等的实现方法, 在条件允许的情况下, 对这些单元进行拟态化处理或冗余处理, 可以减小单点失效的可能性。最终拟态防御系统中是存在单点失效部件的, 需要进行特别防护, 可以借助已有的成熟技术进行保护。拟态防御不排斥已有安全技术手段。

虽然存在单点失效的风险, 但拟态防御将系统的安全问题缩小到了可控的范围内, 将系统整体的安全风险集中到了单点、局部范围内, 降低系统外在风险, 达到防御的目的。

5 总结与展望

本文介绍了软硬件多样性以及基于软硬件多样性的网络空间主动防御技术, 主要对比分析了基于多样性的入侵容忍、移动目标防御和拟态防御的技术选择、实现架构和防御效果等多个层面的优缺点, 为主动防御技术的进一步研究和发展提供参考。

主动防御技术仍有待进一步的发展和研究。多样性的安全效益, 作为主动防御技术的基本出发点, 其安全效益大小对主动防御技术的防御效果评估有着深刻的影响, 同时也是主动防御技术的科学性依据来源。另外, 效能与安全性的折衷始终是网络空间安全技术的焦点, 如何在尽可能低的性能损失、资源消耗的情况下, 获得较高的安全等级, 是主动防御技术发展所必须要解决的关键问题。

在网络空间安全技术的发展历程中, 各类安全技术的产生都是网络攻防博弈的技术结晶, 通过汲取和重组已有技术, 可以实现更顺应网络空间安全新形势的安全技术。主动防御与被动防御技术在本质上并不矛盾, 新型的主动防御技术应当兼容并有效利用已有安全技术, 从全局层面上设计和研究相关技术架构、体系和体制机制, 从而逐渐改变攻防博弈的不对称现状, 以便主动掌握、安全开发网络空间。

参考文献

[1] P.S. Kenkre, A. Pai and L. Colaco, “Real time intrusion detection and prevention system,” *In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, pp. 405-411, 2015.

[2] J. E. Just and M. Cornwell, “Review and analysis of synthetic diversity for breaking monocultures,” *In Proceedings of the 2004*

ACM workshop on Rapid malcode, pp. 23-32, 2004.

[3] B. Baudry and M. Monperrus, (2015). "The multiple facets of software diversity: Recent developments in year 2000 and beyond," *Eprint Arxiv*, vol. 48, no.1, pp.1-26, 2014.

[4] Y.J. Zhao, Z. Y. Tang, N. Wang, D. Y. Fang and Y.X. Gu, "Evaluation of code obfuscating transformation". *Ruanjian Xuebao/Journal of Software*, vol. 23, no.3, pp. 700-711, 2012. (赵玉洁, 汤战勇, 王妮, 房鼎益, 顾元祥, "代码混淆算法有效性评估", *软件学报*, 2012,23(3):700-711。)

[5] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal and M.Franz, "Moving Target Defense, " Springer New York,2011.

[6] M. Franz, "E unibus pluram: massive-scale software diversity as a defense mechanism," *In Proceedings of the 2010 workshop on New security paradigms*, pp. 7-16, 2010.

[7] A. Avizienis and L. Chen, "On the implementation of N-version programming for software fault tolerance during execution," *In Proc. IEEE COMPSAC*, pp. 149-155,1977.

[8] J.X. Wu, "Cyberspace Mimic Defense". Technical report, National Digital Switching System Engineering & Technological R&D Center, 2015. (邬江兴, "网络空间拟态防御", 技术报告, 国家数字交换系统工程技术研究中心, 2015。)

[9] Lardner D." Babbage's calculating engine," *Edinburgh Review*, vol. 59, no. 120, pp. 263–327, 1834.

[10] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, vol.33, no.1, pp.31–88, 2001.

[11] P. R. Lorczak, A. K. Caglayan, and D. E. Eckhardt, "A Theoretical Investigation of Generalized Voters for Redundant Systems," *in The Nineteenth International Symposium on Fault-Tolerant Computing*, pp. 444 – 451, 1989.

[12] Y.W. Leung, "Maximum likelihood voting for fault-tolerant software with finite output-space," *IEEE Trans. Reliability*, vol.44, no.3, pp.419–427, 1995.

[13] G. Latif-Shabgahi, J.M. Bass and S. Bennett, "History-based weighted average voter: a novel software voting algorithm for fault-tolerant computer systems" in *Proceedings. Of Parallel and Distributed Processing, Ninth Euromicro Workshop on. IEEE*, pp.402-409, 2001

[14] G. Latif-Shabgahi and S. Bennett. "Adaptive majority voter: a novel voting algorithm for real-time fault-tolerant control systems," *EUROMICRO Conference, 1999. Proceedings*, pp.113-120, 1999.

[15] F.B. Schneider, "Implementing fault-tolerant services using the state machine approach: a tutorial," *ACM Computing Surveys*, vol.22, no. 4, pp. 299-319, 1990.

[16] L. Gkatzikis and I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems," *IEEE Wireless Communications*, vol.20, no.3, pp.24–32, 2013.

[17] Partha Pal et al., "An architecture for adaptive intrusion-tolerant applications". Software: Practice and Experience, vol. 36, pp. 1331-1354, 2006.

[18] Y. Huang and A.K. Ghosh, "Introducing diversity and uncertainty to create moving attack surfaces for web services," *Moving Target Defense*, Springer New York, pp. 131-151, 2011.

[19] A.K. Bangalore and A.K. Sood, "Securing web servers using self cleansing intrusion tolerance (scit)," *DEPEND'09*, Second International Conference on. IEEE, pp. 60-65, 2009.

[20] H. Okhravi, A. Comella, E. Robinson, et al., "Creating a cyber moving target for critical infrastructure applications using platform diversity," *International Journal of Critical Infrastructure Protection*, vol.3,no.1, pp. 30-39, 2012.

[21] J. Jatzkowski and B. Kleinjohann, "Self-reconfiguration of real-time communication in cyber-physical systems," *Mechatronics*, vol.34, pp.72–77, 2016.

[22] F. Wang, R. Uppalli and C. Killian, "Analysis of techniques for building intrusion tolerant server systems" *Military Communications Conference, 2003. MILCOM*, pp.729-734, 2003.

[23] D. Powell and R. Stroud, "Conceptual model and architecture of MAFTIA," *Technical Report Series-University of Newcastle Upon Tyne Computing Science*, 2003.

[24] M. Al-Kuwaiti, N. Kyriakopoulos and S. Hussein, "A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability," *Communications Surveys & Tutorials*, IEEE, vol. 11, no. 2, pp. 106-124, 2009..

[25] P. Pal, F. Webber, R.E. Schantz., et al., "Intrusion tolerant systems," *in Proceedings of the IEEE Information Survivability Workshop (ISW-2000)*, pp. 24-26, 2000.

[26] V. Gupta, V. Lam, H.G.V. Ramasamy, et al., "Dependability and performance evaluation of intrusion-tolerant server architectures," *Dependable Computing, Springer Berlin Heidelberg*, pp. 81-101, 2003.

[27] F. Wang, F. Jou, F. Gong, et al., "SITAR: A scalable intrusion-tolerant architecture for distributed services", *IEEE Workshop on Information Assurance and Security*, vol.1, pp.1100, 2003.

[28] D. Malkhi and M. Reiter, "Byzantine quorum systems," *Distributed Computing*, vol. 11, no. 4, pp. 203-213, 1998.

[29] L. Zhou, F.B. Schneider. and R. Van Renesse, "COCA: A secure distributed online certification authority," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 329-368, 2002.

[30] G. Tally, B. Whitmore, D. Sames, et al., "Intrusion tolerant distributed object systems: project summary", in *Proceedings of DARPA Information Survivability Conference and Exposition*, vol. 2, pp. 149-151, 2003.

[31] J.H. Lala, "Organically Assured & Survivable Information Systems (OASIS): Foundations of Intrusion Tolerant Systems," *IEEE Computer Society Press*, 2003.

[32] T.P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Advance in Cryptology(CRYPTO'91)*, pp. 129-140, 1994.

[33] M.,A. Marsh and F.B. Schneider, "CODEX: A robust and secure secret distribution system," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 1, no. 1, pp. 34-37, 2004.

[34] M. Malkin, T. Wu and D. Boneh. "Building intrusion tolerant applications," *In Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX)*, IEEE, vol. 1, pp. 74-87, 2000.

[35] D. Obrien, R. Smith, T. Kappel, et al., "Intrusion tolerance via network layer controls," in *Proceedings of DARPA Information Survivability Conference and Exposition*, vol.1, pp. 90-96, 2003.

[36] P. Sousa, A.N. Bessani and R.R. Obelheiro, "The FOREVER service for fault/intrusion removal,".in *Proceedings of the Work-*

shop on Recent Advances on Intrusion-Tolerant Systems, pp. 1-6, 2008.

[37] J.C. Knight, E.A. Strunk and K.J. Sullivan, "Towards a rigorous definition of information system survivability," in *Proceedings of DARPA Information Survivability Conference and Exposition*, pp. 78-89, 2003.

[38] D.L. Kewley and J.F. Bouchard, "DARPA Information Assurance Program dynamic defense experiment summary," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 4, pp. 331-336, 2001.

[39] H. Okhravi, T. Hobson, D. Bigelow, et al., "Finding focus in the blur of moving-target techniques," *Security & Privacy*, vol. 12, no. 2, pp. 16-26, 2014.

[40] M. Dunlop, S. Groat, W. Urbanski, et al., "Mt6d: A moving target ipv6 defense," *2011 Military Communications Conference (MILCOM),* pp.1321-1326, 2011.

[41] J. Knight, D. Heimbigner and A. Wolf, "The willow architecture: Comprehensive survivability for large-scale distributed applications," *in Intrusion Tolerant System Workshop*, Supplemental Volume on 2002 International Conference on Dependable System and Networks, pp. C–7–1, 2002.

[42] J.H. Jafarian, E. Al-Shaer and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in *Proceedings of the first workshop on Hot topics in software defined networks*, ACM, pp.127-132, 2012.

[43] J. Xu, P. Guo, M. Zhao, et al, "Comparing different moving target defense techniques," *In Proceedings of the First ACM Workshop on Moving Target Defense*. ACM, pp. 97-107, 2014.

[44] Q.L. Nguyen and A. Sood, "Improving Security Level via Velocity of Moving Target Defense", *Software Quality, Reliability and Security Companion (QRS-C)*, 2016 IEEE International Conference on. IEEE, pp. 418-419, 2016.

[45] G. Cai, B. Wang, Y. Luo, et al, "Characterizing the running patterns of moving target defense mechanisms," *2016 18th International Conference on Advanced Communication Technology (ICACT)*. IEEE, pp.191-196, 2016.

[46] M. Thompson, N. Evans and V. Kisekka, "Multiple os rotational environment an implemented moving target defense," *Resilient Control Systems (ISRCS)*, 2014 7th International Symposium on. IEEE, pp. 1-6, 2014.

[47] W. Peng, F. Li, C.T. Huang, et al., "A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces", *2014 IEEE International Conference on Communications (ICC)*, IEEE, pp.804-809, 2014.

[48] C. Hewa Nadungodage, Y. Xia, P.S. Vaidya, et al., "Online multi-dimensional regression analysis on concept-drifting data streams," *International Journal of Data Mining*, vol. 6, no. 3, pp.217-238, 2014.

[49] G.L. Cai, B.S. Wang, T.Z. Wang, et al, "Research and development of moving target defense technology," *Journal of Computer Research and Development*, vol. 53, no. 5, pp. 968-987(in Chinese), 2016.
(蔡桂林, 王宝生, 王天佐等, "移动目标防御技术研究进展", *计算机研究与发展*, 2016, 53(5): 968-987。)

[50] J.X. Wu, "Meaning and Vision of Mimic Computing and Mimic Security Defense," *Telecommunications Science*, vol. 30, no. 7, pp. 1-7(in Chinese), 2014.
(邬江兴, "拟态计算与拟态安全防御的原意和愿景", *电信科学*, 2014, 30(7): 1-7。)

[51] J.X. Wu, "Cyber space Mimic Security Defense," *Secrecy Science and Technology*, vol. 10, no. 1, pp. 4-9(in Chinese), 2014.
(邬江兴, "网络空间拟态安全防御", *保密科学技术*, 2014, 10(1): 4-9。)

[52] J.X. Wu, F. Zhang and X.G. Luo, "Mimic computing and Mimic Security Defense," *Communications of the CCF*, vol. 11, no. 1, pp. 8-14(in Chinese), 2015.
(邬江兴, 张帆, 罗兴国, "拟态计算与拟态安全防御", *中国计算机学会通讯*, 2015, 11(1): 8-14。)

[53] "The prototype of cyberspace mimic defense in routing system", Technical report, National Digital Switching System Engineering & Technological R&D Center, 2015.
("路由器拟态防御原理验证系统", 技术报告, 国家数字交换系统工程技术研究中心, 2015。)

[54] "The prototype of cyberspace mimic defense in web servers", Technical report, State Key Laboratory of Mathematical Engineering and Advanced Computing, 2015.
("web 服务器拟态防御原理验证系统", 技术报告, 数学工程与先进计算国家重点实验室, 2015。)

仝青 于 2014 年在解放军信息工程大学计算机科学与技术专业获得学士学位。现在解放军信息工程大学计算机科学与技术专业攻读硕士学位。研究领域为网络安全。研究兴趣包括: 主动防御技术。Email: szbnlllskd@163.com

张铮 于2006年在解放军信息工程大学计算机科学与技术专业获得博士学位。现任数学工程与先进计算国家重点实验室副教授。研究领域为网络安全、先进计算。研究兴趣包括: 主动防御技术、高性能计算。Email: ponyzhang@126.com

邬江兴 现任国家数字交换系统工程技术研究中心主任, 教授, 博导。研究领域为信息通信网络、网络安全。Email: 17034203@qq.com